大数据心理学的 Python 入门

吕思华 [1,2] 宋梦瑶 [1,2] 莫柳铃 [1,2] 李东启 [1,2] 朱廷劭 [1,2*]
[1]（中国科学院大学心理学系，北京 100049）
[2]（中国科学院心理研究所，北京 100101）

摘要 ：

本文以九九文章网为例，详细地介绍了大数据心理学研究方法。利用用户实验采集的文本数据，提取词频特征，训练机器学习模型，然后利用学习模型实现对爬取的九九文章网的文章对应的生活满意度进行预测，帮助大数据研究初学者对整个处理流程有直观的感受。本文通过具体实例，介绍了 Python 和情感词典用于文本的词频计算，利用 scikit-learn 库完成对机器学习模型训练、测试及应用，并结合附带的源程序，便于读者直接操作。本文初步介绍了基于文本词频的机器学习建模的大数据研究方法，对于其中技术的介绍较为基础，主要强调如何将技术进行应用，对技术原理的介绍较少。

关键词： 大数据，词频，机器学习，python

Python for Big Data Psychology Research

Lyu Sihua[1,2] Song Mengyao[1,2] Mo Liuling[1,2] Li Dongqi[1,2]
Zhu Tingshao[1,2*]
[1] (Department of Psychology, University of Chinese Academy of Sciences, Beijing 100049)
[2] (Institute of Psychology, Chinese Academy of Sciences, Beijing 100101)

Abstract:

This paper introduces the big data research method in psychology in details, taking Ninety-Nine Articles website as an example. Using the collected textual data, we calculated word frequencies as features, then trained machine learning models, and used models to predict life satisfaction for texts crawled from Ninety-Nine Articles website, providing inspiration and help for beginners in big data research. This paper introduces text-based word frequency calculation using Python and sentiment dictionary through specific examples, and completes the training, testing and application of the machine learning model using Python's scikit-learn library. Furthermore, we uploaded the accompanying source program for direct operation. This paper introduces the big data research method of machine learning modeling via text-based word frequency. Our article emphasizes how to apply the technology, and thus we introduce the technology in a more basic way with less involvement of the technical principles.

Keywords: big data, word frequency, machine learning, python

1. 引言

随着信息技术的不断发展，互联网使用过程中衍生出的海量数据成为最具价值的财富。各种数据渗透着人们的生活，并以指数级的速度在增长，数据爆炸[1]将我们带入大数据时代。大数据开始蔓延到社会的各行各业从而影响着我们的学习、工作、生活以及社会的发展，也为研究人员开展研究带来了很大的便利[2-3]。互联网作为海量数据的最主流载体之一，存在大量没有被有效利用的数据，亟待人们去挖掘其背后的价值。Python 作为最接近自然语言的计算机语言，学习起来相对简单，成为人们进行有效数据挖掘的有力工具。

Python 作为一种面向对象的计算机程序语言，存在着丰富的库和应用程序编程接口(API)，可以满足网络平台中多元化数据信息的挖掘、分析需求，能够实现对海量数据信息的查找、保存与显示[4]。嵩天、黄天羽等人认为相对于传统 c++，vb 等教学语言，Python 更适合非计算机专业的学生，教学的应用场景更广[6]。郑戟明认为 Python 具有简单易学、丰富的类库，优良的可扩展性和可移植性等特点，非常适合培养学生的计算思维[7]。2021 年 10 月，Python 在 TIOBE 排名榜上超过 C 语言和 Java，荣升第一。

大数据时代背景下，Python 无疑是进行大数据研究的得力助手。基于当下人们掌握 Python 的现实状况，本文将结合实际数据，分四个步骤介绍如何使用 Python 开展大数据心理学研究：Python 基本入门介绍；爬虫网络数据下载；jieba 分词及词频统计；机器学习模型的训练、测试及应用。同时，结合本文附带的源代码和数据，可以帮助相关人员逐步了解开展大数据心理学研究的主要技术，并通过运行相关的程序对研究过程进行实际操作，其中的部分处理程序也可以运用在自己开展的课题研究中。

2. Python 简介

Python 是一门简单，可视化程度高的编程语言，简单易学，功能强大，同时有高效率的高层数据结构，能够简单、有效地实现面向对象编程。Python 具有简洁的语法，支持动态输入，是解释性语言。在大多数平台上，对于众多领域，Python 都是一个理想的开发语言。

Python 是免费开源的，具有很好的可移植性，可以运行在 Unix 衍生系统、Win32 系统、掌上平台（掌上电脑/手机）以及游戏控制台（PSP）等等。

Python 拥有庞大的标准库，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、机器学习等。此外，还有其他高质量的库，如 wxPython、Twisted 和图像库等

人们可以通过 Python 的官方网站进行下载安装：http://www.python.org。一般 Unix 衍生系统可能预安装了 Python，在命令行对话框中键入"python"即可显示版本信息。安装 Python 其实很简单，和安装其他软件类似，并且网上有许多开放的安装教程可供参考。网络上也有很多的 Python 编程学习教程：

W3school 在线教程：https://www.w3school.com.cn/python/index.asp

简明教程: https://www.woodpecker.org.cn/abyteofpython_cn/chinese/

Python3 教程：https://www.runoob.com/python3/python3-tutorial.html

啄木鸟社区：https://wiki.woodpecker.org.cn/moin/

3. Python 编程基础

在 Python 编程中，数据类型是一个重要的概念。变量可以存储不同类型的数据，并且不同类型可以执行不同的操作。Python 中有六个标准的数据类型：Number（数字），String（字符串），List（列表），Tuple（元组），Set（集合），Dictionary（字典）。其中，不可变数据包括：Number（数字）、String（字符串）和 Tuple（元组），可变数据包括 List（列表）、Set（集合）和 Dictionary（字典）。

表 1 Python 的基本数据类型

基本数据类型	包括的种类	说明
Number（数字）	int（整数）	在数字中，正整数、0、负整数都称为整型。
	float（浮点数）	含有小数点的数据都是浮点型。
	bool（布尔型）	只有两种情况：True、False，表示真假。
	complex(复数）	复数为实数加虚数，只要存在虚数，此数据类型就为复数类型，如 3j
String(字符串)	str	由引号括起来的都是字符串，如"123"、'hello'
List(列表)	list	可获取，可修改，有序。正向索引：0、1、2、3……逆向索引：-1、-2、-3、-4、-5……
Tuple（元组）	tuple	可获取、不可修改、有序。索引时的正向与逆向下标与列表相同。
Set(集合)	set	无序、不可修改、自动去重。引时的正向与逆向下标与列表相同。
Dictionary（字典）	dict	以键值对的形式存储数据，通过键来获取和修改值。

使用 type() 函数可以获取任何对象的数据类型，使用 print（）函数可以对结果进行打印，可参见 **prog\simp-1-hello-world.py**。

Python 运算符用于对变量和值执行操作,分为：算术运算符，赋值运算符，比较运算符，逻辑运算符，身份运算符，成员运算符，位运算符。Python 运算符的使用参见 **prog\simp-5-exp.py**。在 Python 编程中，if 语句用于控制程序的执行。用 if 关键字表示判断条件，当判断条件成立时，则执行后面的语句，执行的语句可以多行，以缩进来区分，表示同一个判断条件的执行内容。elif 关键字是对“如果之前的条件不正确，那么试试这个条件”的表达方式。else 关键字可以用来捕获未被之前的条件捕获的所有情况。示例代码参见 **prog\simp-6-ifelse.py**。

```
# 判断成绩
x = 88
print('成绩为：', x)
if(x >= 85):
    print("优")
elif(x >= 75):
    print(" 良")
elif(x >= 60):
    print("中")
else:
    print('差')
```

Python 的 while 语句用于循环执行程序，即在某种条件下，循环执行某段程序，以处理需要重复处理的相同任务。示例代码来自 **prog\simp-7-while.py**。在 while 循环中，只要条件为真，就可以执行一组语句。

```
# 利用循环，打印出 10 以内两个不同数字的组合
i = 0
while i < 10:
    j = i +1
    while j < 10:
        print(i,'：',j)
        j=j+1
    i = i + 1
```

在 While 循环中，使用 break 语句，即使 while 条件为真，也可以终止循环。使用 continue 语句，可以跳出当前这个迭代，然后继续下一个迭代。

for 循环用于迭代序列（即列表，元组，字典，集合或字符串），序列可以从 0 开始索引，以小于序列的长度结束，并且可以被切分。示例代码来自 **prog\simp-9-for.py**。通过使用 for 循环，可以为列表、元组、集合中的每个项目执行相同的一组语句。

```python
#遍历打印 num 列表中所有的数
num= [1, 92, 3, 6, 88, 9, 2, 76, 8, 45, 78, 786, 8, 10]
for x in num:
    print(x)
```

此外，在 for 循环中，使用 break 关键字，可以在循环遍历所有项目之前终止循环。使用 continue 关键字，可以停止循环的当前迭代，并继续下一个迭代。else 关键字在 for 循环中可以指定循环结束时要执行的代码块。

函数是一种仅在调用时运行的代码块，可以将数据（称为参数）传递到函数中，可以在函数名后的括号内指定参数。函数可以把数据作为结果返回。一般要先对函数进行定义才能调用，示例代码来自 **prog\simp-4-function.py**。在 Python 中，使用 def 关键字来定义函数：

```python
# 定义一个相乘的函数
def multiply(a, b, c):
    print('函数输入：')
    print('a = ', a)
    print('b = ', b)
    print('c = ', c)
    z = a * b * c

    print('函数输出：', z)
    return z
```

在函数名称后跟括号来调用函数：

```python
# 直接调用该函数
print('a * b * c = ', multiply(8, 9, 7))
print('a * b * c = ', multiply(5, 4, 70))
```

4. 爬虫网络数据下载

网络爬虫是一种按照一定规则，自动地抓取网页信息的程序，它的基本工作方式是模拟人工的操作。本节主要介绍网络爬虫技术的工作流程，结合实际案例来讲解基于 Python 的网络数据下载过程。

传统爬虫从一个或若干初始网页的 URL 开始，获得初始网页上的 URL，在抓取网页的过程中，不断从当前页面上抽取新的 URL 放入队列,直到满足系统的一定停止条件。如图 1 所示，网络爬虫的基本工作流程如下：（1）首先选取一个或多个 URL 作为种子 URL；（2）将选取的种子 URL 放入到待抓取 URL 中；（3）依次从待抓取 URL 队列中取出 URL，对 URL 的 DNS 进行解析，获得主服务器 IP，并将网页下载下来，保存到数据库中。然后将该 URL

放入已抓取 URL 队列中；（4）分析已抓取 URL 队列中的 URL，得到另一些 URL，再次放入待抓取 URL 队列，从而继续循环下去[5]。

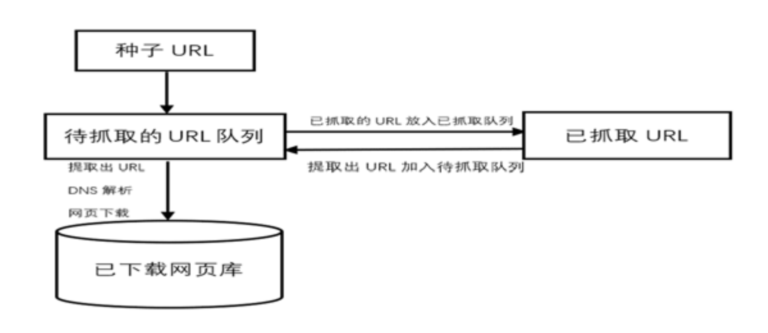


图 1 网络爬虫的基本工作流程

我们这里以九九文章网为例，爬取其中文章的标题及内容，爬取目标网站链接：*http://www.99wenzhangwang.com/article/18491.html*，将文章标题和内容保存在本地 txt 文件中。本案例在 Python3.8 环境下完成，完整代码见 **prog\www-9-99wz-sample.py**，网络数据下载可以归为获取数据、解析数据，提取数据，储存数据四个步骤。

爬虫程序会根据我们提供的网址，向服务器发起请求，然后返回数据。获取数据需要用到 requests 库，requests 库可以下载网页源代码、文本、图片，甚至是音频。由于 requests 库不是 Python 标准库，需要单独安装它。在终端输入：pip install requests（Mac 电脑输入 pip3 install requests），安装完成后即可使用。使用 requests 库中的 get()方法向服务器发送请求，向服务器发出请求后，服务返回的是一个带有 HTML 文档的数据包。具体代码如下：

```
#引入 requests 库，bs4 库
import requests
from bs4 import BeautifulSoup
# 发送请求，并把响应结果赋值在变量 r 上
r = requests.get('http://www.99wenzhangwang.com/article/18491.html')
# 解决中文乱码
r.encoding = r.apparent_encoding
```

HTML 全称为 Hypertext Markup Language(超文本标记语言)，是一种用于创建网页的标准标记语言。标记语言就是把文本和文本以外的相关信息（例如大小，高度，颜色，位置等）组合在一起的语言。如图 2 所示，HTML 和 Python 一样有着明显的层级结构。图中一共有三组基本元素，分别是 html 元素（<html></html>），head 头元素（<head></head>），和 body 主体元素（<body></body>）。头部元素，一般用来设置网页的编码，添加网页标签的

logo，小标题，外部文件引用等。主体元素负责定义网页窗口内的所有内容，是今后我们重点关注的对象。元素中用尖括号（<>和</>）括起来的字母和英文是标签，用于标记文本信息。其中 align 和 style 为属性，用来定义元素的样式。我们一般就是根据标签名和属性值来定位我们想要数据的位置。

```
<!DOCTYPE html>                       #全局声明，这是一个html文档
<html>                                #html文档（开始）
  <head>                                #文档头（开始）
    <title>九九文章网</title>              #文档的标题
  </head>                               #文档头（结束）
  <body>                                #文档体（开始）
     <h1 align='center' style='color: #20b2aa;'>HTML介绍</h1>    #一级标题
     <p>第一段</p>                          #段落
  </body>                               #文档体（结束）
</html>                               #html文档（结束）
```

图 2　HTML 文档结构

解析网页有正则表达式、BeautifulSoup、 lxml 等多种方式，每种方式各有特色，可结合实际进行选择。本案例中使用 BeautifulSoup 对网页进行解析。由于 BeautifulSoup 4 库不是 Python 标准库，我们需要单独安装它，安装方法同上。BeautifulSoup 解析数据时需要输入两个参数，第一个参数为要解析的文本，必须为字符串类型。第二个参数用来标识解析器，我们用的是一个 Python 内置库：html.parser。具体代码如下：

```
# 把网页解析为 BeautifulSoup 对象
soup = BeautifulSoup(r.text,'html.parser')
```

将获取的网页解析为 BeautifulSoup 对象后，需要通过标签和属性来获取想要的数据位置。这里通过在 Chrome 浏览器中安装 SelectorGadget 插件确定标签，标签确定方法如图 3，插件安装和使用见 **software\chrome\install.pdf**。标签确定后，使用 find()和 find_all()方法来提取数据。find()方法返回符合要求的首个数据；find_all()方法以列表形式返回所有符合要求的数据，需通过遍历列表提取所需数据，具体代码如下：

```
# 用 find()把符合要求的首个数据提取出来,并赋值给变量 title
title = soup.find('h2')
# 用 find_all()把符合要求的所有数据提取出来，并赋值给变量 contents
contents=soup.find(class_='hl_body').find_all('p')
#定义一个空列表
content=""
#遍历 contents 列表，提取列表中的文字并赋值给 content
for para in contents:
    if len(para)>0:
        content += para.text
```



图 3 标签确定方法

在所需数据提取出来后，需要储存数据。数据存储的方式有多种，可以保留在普通文件中，如 txt、csv，Excel 等；也可以存储在数据库中，如 MySQL。本案例中将爬取到的数据写入 txt 文件中。文件保存需要三步：打开文件—写入文件—关闭文件。具体代码如下：

```
#打开文件
file = open('99wenzhang.txt','w',encoding='utf8')
#写入文件
file.write(title.text)
file.write('\n')
file.write(content)
#关闭文件
file.close()
```

数据保存结果如图 4：

人生 经历了才深深懂得 人生的起起落落 总会有一些情怀需要安人生

你要错过很多人，才知道把握对的人。要喝过很多不同种类的饮料，才会懂得回归白开水的平淡。要司空见惯城市的喧嚣，才会想念田园的安静。要买过很多衣服，才知道哪款才是经典。突然想到有个电

人生，经历了才深深懂得，人生的起起落落，总会有一些情怀需要安静回味，总会有一些经历需要独自体会，总会有一段路需要一个人走，总会有一些事需要坦然面对。轻捻滑落指尖的光阴，拥有那些静谧的时光，
放下尘世纷纷扰扰，不计较过去，不执着未来，不纠结当下，得到的，失去的，付出的，收获的，都是成全人生的章章节节，拥有的就万分珍惜，错过的，就看做人世尘烟，不留恋惋惜，一切顺其自然，让

不哭，不哭，为了自己加油，别愁，别愁，为了自己抬头。你的走，我不能留，我只能给自己加油，为自己抬头，我不会因你的走而颓废，因为我是情场高手，但高手也会泪流。感伤，哪有？凄凉，哪有？
在生命里出现的每个人，都有原因，喜欢你的人给了你温暖和勇气，你喜欢的人让你学会了爱和自持，你不喜欢的人教会了你宽容尊重，不喜欢你的人让你知道了自省成长，没有人会无缘无故出现在你生命

有些东西，无论别人说的再好，说的再坏，只有在你自己亲身经历之后，才知道它是好的？还是坏的？就像咖啡一样，是苦是甜，只有喝的人才会知道，而我们永远只是沉浸在别人的故事里，留自己的眼泪
（来源：网络）

图 4　保存结果示例

依据同样方式，可以建立特定网站的网络爬虫，**prog\www-1~7-*.py** 的 7 个 Python 程序逐步介绍了针对中科院心理所通知公告栏的爬虫实现，**prog\www-8-99wzsave.py** 则是针对九九文章网的某一特定栏目的文章下载。

5. jieba 分词及词频统计

文本数据进行机器学习涉及到的关键问题之一是如何得到可作为输入数据的文本数据特征。原始的文本数据一般情况下被表示由字符组成的字符串，首先，机器学习模型一般无法直接对字符类型的数据进行处理；其次，不同的文本数据字符长度往往不同，而机器学习模型一般要求输入数据具有相同的维度。因此，在对于文本数据进行机器学习模型建模时，需要首先对原始文本数据进行处理，得到可以用数值表征的文本数据特征，其主要处理技术和相关流程如图 5 所示。

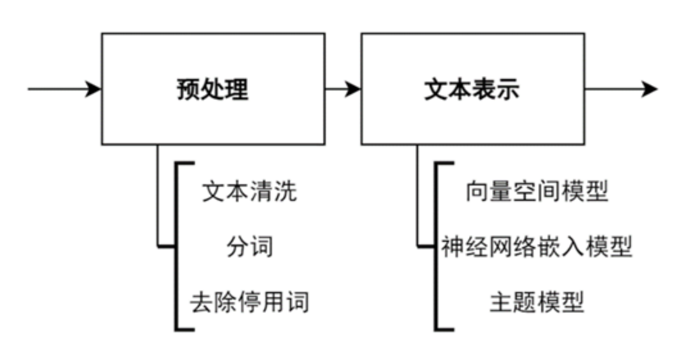


图 5 文本数据特征获取主要技术及相关流程

在将原始文本数据转换为数值表征的文本数据特征的过程中，首先要对原始文本数据进行文本预处理。预处理的第一步是对于原始文本进行文本清洗。由于原始文本数据很多时候具有空格或者是其他一些无用的符号，如果保留这些符号，分词结果中也会体现这些符号，一方面这些符号对于词频统计过程产生干扰，另一方面这些无用符号并不是我们所需要的文本数据特征。因此需要对无用符号进行替换或者删除，这个过程被称为文本清洗。

其次是分词过程，分词就是将连续的字符序列按照一定的规范重新组合成语义独立词序列的过程。英文行文中，单词之间以空格作为自然分界符，中文的词没有形式上的分界符，对中文文本进行分词具有一定的难度。但目前已有多种技术和软件实现了对于中文文本进行分词，本文将要介绍的基于 Python 编程语言的 Jieba 便是其中一种分词工具，参见 **prog\jieba-1-seg.py**。

文本数据处理的最后一步是去除停用词(**prog\jieba-2-stopwords.py**)。停用词可以简单地被理解为不被纳入词频统计的词语。在中文文本处理中，此类词一般为介词、助词或者副词等，停用词表根据词频统计任务目的的不同而不同。

对于文本数据进行预处理后，需要将文本数据转换为数值类型，这一过程被称为文本表示。文本表示过程主要有三大主要模型，分别是向量空间模型、神经网络嵌入模型和主题模型。向量空间模型将文本数据表示为向量空间中的点，其关键在于特征向量的选取和特征向量的权值计算两个部分。神经网络浅入模型采用深度学习的方法，实现离散的文本数据变量到连续数字向量的映射。主题模型以无监督学习的方式对文本的隐含语义结构进行聚类的统计学模型。词频统计过程可以被认为是向量空间模型的一种简化模式，它通过预先设定的词典文件对于文本数据中不同词类出现的频率进行统计分析，得到每个文本的词类频率向量，这些词类频率向量可以直接用于后续的监督学习过程或者统计分析。

本节以通过九九文章网爬取的部分文章作为原始文本数据，以大连理工情感词典和微博客基本情绪词库为词典文件对于文本进行分词处理和词频统计。其中，大连理工情感词典包含快乐、安心、尊敬等共 21 个词类，微博客基本情绪词库包含快乐、悲伤、愤怒、恐惧和厌恶共 5 个词类。首先对于不同词类的标识进行定义，本节中的示例代码参见 **prog\swls-2-jieba-affect-export.py**：

```
affect_col_list = ['PA', 'PE','PD','PH','PG','PB','PK',
                   'NA','NB','NJ','NH','PF','NI','NC',
                   'NG','NE','ND','NN','NK','NL','PC',
                   'MH', 'MS', 'MA', 'MD', 'ME',
                   'P','N','Ne']
```

在使用 Jieba 进行分词和词频统计之前，首先要对词典文件和停用词表进行载入，载入情感词典的步骤如下：

```
def load_affect_dict(filepath):
    m_affectdict = []
    for m_col in affect_col_list:
        m_col = []
        m_affectdict.append(m_col)
    for m_line in open(filepath, 'r', encoding='utf-8').readlines():
        m_line=m_line.strip()
        kwd = m_line.split('\t')[0].strip()
        col = m_line.split('\t')[1].strip()
        m_affectdict[affect_col_list.index(col)].append(kwd)
    return m_affectdict

#载入情感词典
affect_dict_file = '../data/dict-affect.txt'
affect_dict = load_affect_dict(affect_dict_file)

#载入情感词典中的词做为自定义词典
jieba.load_userdict('../data/jiebaload_affect_dict.txt')
```

载入停用词表的步骤如下：

```
# 创建停用词 list
def load_stopwords(filepath):
    m_stopwords = [line.strip() for line in open(filepath, 'r', encoding='utf-8').readlines()]
    return m_stopwords
#载入停词表
stop_word_file = '../data/stop_words_cn.txt'
stopwords = load_stopwords(stop_word_file)
```

载入中文停用词表和大连理工情感词典以及微博客基本情绪库后，可以使用 jieba 对于文本进行分词和去停词处理，并对于不同词类的词频进行统计，最终得到文本中情感词典各个词类的词频。具体代码如下：

```
#读入一个数据文件，返回得分、性别和自我描述
def read_swls_file(fname):
    #print(fname)
    fr_swls = open(fname, 'r', encoding='UTF-8-sig')
    x_swls_strs = fr_swls.readlines()
     fr_score = int(x_swls_strs[0].strip('\n'))
    fr_gender = x_swls_strs[1].strip('\n')
     fr_desc = ''
    #删除得分和性别行
    x_swls_strs.pop(0)
    x_swls_strs.pop(0)
    for swls_str in x_swls_strs:
        fr_desc += swls_str.strip().strip('\n') + ' '
    fr_swls.close()
    return fr_score, fr_gender, fr_desc
# 对每个用户的自述文件进行处理，统计各个情绪分类的比率
for fdata in swls_files:
    print(fdata)
    str_export = ''

    x_score, x_gender, x_desc = read_swls_file(swls_dir+fdata)
    str_export += str(x_score)+','

    #男性为 0，女性为 1
    if(x_gender == 'M') or (x_gender == 'm') or (x_gender == '男'):
        str_export += '0'
    else:
        str_export += '1'

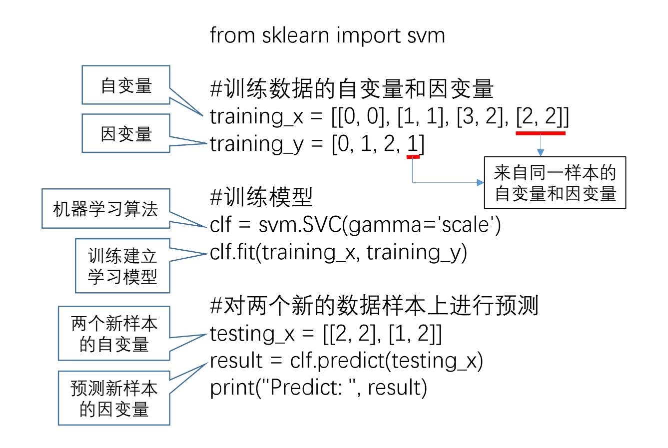
    idx = 0
    for g_col in affect_col_list:
        r_affect = cntkws_jieba_seg_wrds(x_desc, affect_dict[idx])
        str_export += ',' + str(r_affect)
        idx += 1

    dstfp.write(str_export)
    dstfp.write('\n')
    dstfp.flush()
dstfp.close()
```

对于每个文本而言，文本数据都被转换为了具有相同维度的数值类型的向量，并且每个维度具备一定的语言学意义，因此可以进一步进行监督学习过程或者直接进行统计分析。

6. 机器学习模型的训练、测试及应用

机器学习方法能够从大量数据中挖掘出隐含的规律，并依据此规律对未来的输入数据进行预测识别。机器学习模型分为有监督学习和无监督学习(**prog\ml-1-kmeans.py**)，常见的有监督学习模型包括分类(**prog\ml-5-svm.py**)和回归模型(**prog\ml-4-svr.py**)。下面我们以 **prog\ml-5-svm.py** 为例，介绍机器学习模型的训练和应用的主要过程。



一般来说，为了能够对新样本进行自动预测，我们需要首先训练得到一个预测模型，然后才可以利用预测模型进行预测。为了训练得到预测模型，我们需要获取数据，每条数据是针对一个样本，包括自变量和因变量。自变量一般包括多个，有的时候，也把一个自变量叫一个特征，这样自变量部分如果包括多个自变量的话，也被称为特征向量，因变量有的时候也称为标注或者输出。训练模型的过程，就是让计算机自动学习到自变量和因变量之间的映射关系，这样在应用的时候，我们只输入新样本的自变量，就可以调用训练得到的模型，自动预测出该新样本的因变量，也就是上面展示的“训练”和“预测”。

上面的例子只是简单介绍了训练和预测的过程，一般有监督的机器学习实践过程往往都包括了模型训练、测试、保存和导入应用四大环节。模型训练和测试主要是利用数据在不同的机器学习算法上进行调试，寻找到最佳的预测性能的机器学习算法及其参数设定。一般在这个阶段，从原始数据开始进行特征提取、特征选取、训练建模以及性能测试。整个过程是不断往返的，也就是

说，根据性能测试的结果，可以回溯到前面三个步骤中的任何一个步骤进行调整。为了对模型的性能进行测试，一般把所有数据随机分为训练集和测试集，它们完全分开没有重叠。在训练集上进行模型的学习训练，之后在测试集上对模型的预测性能进行测试。在测试的结果上，确定合适的算法和参数设定，然后在全部的数据集上建立好模型并保存，以便后续导入应用。类似地，对于已经建立好的模型，也可以通过导入模型进行应用，对于后续任意一个没有标注的数据样本，只要能够提取出模型要求的输入特征，都能利用现有模型实现对标注的预测。

下面以生活满意度预测模型为例，说明模型的训练和测试过程。为了建立生活满意度预测模型，我们需要获得每个被试介绍自我目前情况的文本以及生活满意度得分（通过填写生活满意度量表），数据的采集过程参见 **data\生活满意度练习.pdf**。我们可以把每个人的自我描述的那段文本看做是输入的自变量，而通过问卷测量得到的生活满意度得分作为因变量，希望能够建立一个利用个体自我描述的文本实现对其生活满意度的自动预测，

通过前文提到的文本词频统计方法，我们能够从每个被试的自我描述的文本中提取词频特征作为自变量，而生活满意度得分作为因变量。**prog\swls-5-train-save.py** 利用了情感词典来提取个体的词频特征，具体提取过程可参考第五节关于使用 Jieba 进行分词和词频统计的部分。在此基础上，在程序中将个体的词频特征作为输入变量赋值给了 x_kws，个体的生活满意度作为预测变量赋值给 y_score，具体代码片段如下所示：

```
#特征提取并训练模型
x_kws = []
y_score = []

dirs = '../data/swls/'
subdir = os.listdir(dirs)
for f in subdir: # 遍历文件夹下的文件
    print('.', end='')
    x_score, x_gender, x_desc = read_swls_file(dirs+f)
    item = feature_extraction(x_desc)
    x_kws.append(item )
    y_score.append((x_score-5)/30)
```

Python 的 scikit-learn 库是一个开源的机器学习模块（https://scikit-learn.org.cn/），它具有各种常见的分类、回归和聚类算法，可以直接从该库中调用对应的函数来训练机器学习模型，降低了对建模人员的数学能力和计算机能

力的要求。利用 sklearn 进行模型训练的代码如下所示分为两步，第一步指定使用的模型，第二步用训练集数据拟合模型，调用岭回归模型如下所示。

```
#训练模型
clf_lasso = LassoCV()
clf_lasso.fit(x_kws, y_score)
```

以上例举的是基础的模型训练过程，但是对于同一个数据集而言，不同的算法模型，甚至是同一个模型设置的不同超参数也会导致模型性能的差异。因此，还需要进一步利用测试集的方法来确定模型和超参数的选择。

测试集是由一批已经完成标注的数据组成的，既有模型要求的输入数据，也有拟预测对象的真实值，模型可以基于输入数据获得对应的预测值。预测值与真值之间的相关系数可以作为该模型的预测性能的表现之一，相关系数越大则模型性能越好。在生活满意度项目中，**prog\swls-3-train-test-score.py** 文件实现了对于模型的测试。同样以岭回归为例，在该程序中先基于训练集建立了岭回归模型，再基于测试集中个体文本的词频特征，实现对个体的生活满意度的预测，进一步计算出预测值与真值之间的相关系数作为测试结果，代码实现部分如下所示。

```
clf_lasso = LassoCV()
clf_lasso.fit(training_x, training_y)
result = clf_lasso.predict(testing_x)
ab = np.array([testing_y, result])
print('Lasso: ', np.corrcoef(ab))
```

通过比较具有不同超参数的模型在测试集上的效果，我们可以选择出其中表现得最好的模型，利用对应的模型和设置的参数在全部数据上训练出最终模型，导出成模型文件，以方便后续的应用。在 **prog\swls-5-train-save.py** 程序中代码如下所示，以便后续的调用，实现对个体生活满意度的预测。

```
#保存训练得到的模型
mod_file = '../data/swls.mod'
joblib.dump(clf_lasso, mod_file)
print('SWLS model saved!')
```

在建立好模型并导出模型文件后，就可以在新获得个体文本表达的基础上对其生活满意度进行预测，此处我们以从 99 文章网下载下来的文本为例，对文本进行特征提取和生活满意度预测，实现模型的应用。在完成了对各文本相同的词频特征提取工作后，需要加载事先保存下来的模型，然后以词频特征为输

入，对生活满意度进行预测，并输出得分，实现过程在 **prog\swls-6-99wz-apply.py** 中。特征提取和得分预测的具体代码如下。

```
#特征提取代码
testdirs = '../data/99wz/'
testsubdir = os.listdir(testdirs)
for f in testsubdir: # 遍历文件夹下的文件
    print(testdirs+f)
    buf = open(testdirs+f, 'r', encoding='utf-8').read()
    item = feature_extraction(buf)
    #print(item)
    apply_kws.append(item )
    wz_list.append(f)

#得分预测代码
mod_file = '../data/swls.mod'
clf = joblib.load(mod_file)
result = clf.predict(apply_kws)
print(result)
```

7. 总结

本文简要介绍了利用 Python 开展大数据心理学研究的主要过程，并以九九文章网的生活满意度自动预测为例，介绍了基于文本词频的机器学习建模的全流程。结合本文介绍的技术，可以实现对网络用户的熬夜[2]以及出柜与否[3]的研究。研究者可以利用爬虫技术实现对网络数据的定向爬取，对于文本数据，可以结合使用 jieba 分词以及现有词典实现对于文本数据的特征提取；对于有标注的文本数据，可以进一步通过 Python 利用有监督的机器学习算法建立模型并保存，以便未来在相关研究当中可以直接调用，实现对心理指标的自动预测，并在此基础上开展更多的心理学研究。

参考文献

[1] Rupa Mahanti.Data Governance and Data Management[M]. Strathfield: Springer,p1-3.

[2] 宋梦瑶,靳媛媛,柏冰玉,徐颖,马岩,丁晓庆,朱廷劭,赵楠. 熬夜、城市发展水平与生活满意度关系——基于微博大数据的研究[DB/OL]. (2021-12-29) [2022-03-05]. http://www.chinaxiv.org/abs/202112.00163

[3] 王薪舒,赵梦晗,潘超,赵楠,朱廷劭. 出柜与否对于性少数群体心理的影响——基于微博知乎数据的研究[DB/OL]. (2021-12-18) [2022-03-05]. http://www.chinaxiv.org/abs/202112.00120

[4]李玉香,王孟玉,涂宇晰.基于 python 的网络爬虫技术研究[J].信息技术与信息化,2019,(12):143-145

[5]嵩天,黄天羽,礼欣.Python 语言:程序设计课程教学改革的理想选择[J].中国大学教学,2016,(2):42-47.

[6]郑戟明.Python 程序设计课程中计算思维的应用[J].大学教育,2016,(8):127-129.

[7]张艳,吴玉全. 基于 Python 的网络数据爬虫程序设计[J]. 电脑编程技巧与维护,2020(4):26-27. DOI:10.3969/j.issn.1006-4052.2020.04.010.

[8] Tianguiyuyu. K 折交叉验证[DB/OL]. (2018-06-14) [2022-03-05]. https://blog.csdn.net/tianguiyuyu/article/details/80697223